Similarity Detection for Free-Form Parametric Models

Quoc-Viet Dang
University of Toulouse,
France
qdang2@n7.fr

Sandrine Mouysset
University of Toulouse,
France
sandrine.mouysset@irit.fr

Géraldine Morin
University of Toulouse,
France
morin@n7.fr

Abstract

In this article, we propose a framework for detecting local similarities in free-form parametric models, in particular on B-Splines or NURBS based B-reps: patches similar up to an approximated isometry are identified. Many recent articles have tackled similarity detection on 3D objects, in particular on 3D meshes. The parametric B-splines, or NURBS models are standard in the CAD (Computer Aided Design) industry, and similarity detection opens the door to interesting applications in this domain, such as model editing, objects comparison or efficient coding. Our contributions are twofold: we adapt the current technique called votes transformation space for parametric surfaces and we improve the identification of isometries. First, an orientation technique independent of the parameterization permits to identify direct versus indirect transformations. Second, the validation step is generalized to extend to the whole B-rep. Then, by classifying the isometries according to their fixed points, we simplify the clustering step. We also apply an unsupervised spectral clustering method which improves the results but also automatically estimates the number of clusters.

Keywords

similarity detection, parametric surfaces, isometry, spectral clustering

1 INTRODUCTION

Parametric surfaces, in particular Non-Uniform Rational B-Spline (NURBS), provide a powerful tool in the hands of the academic and industrial communities concerned with the design and analysis of objects [Dim99a]. NURBS based B-reps (Boundary representations) are industrial standards and are widely used in different domains such as molecular chemistry [Baj97a], 3D geographical information systems [Cau03a] and mechanical components design [Chu06a]. Additionally, similarity within a 3D shape is a common phenomenon both in natural and in synthetic objects. Many objects are composed by similar parts up to a rotation, a translation or a reflection. Geometric redundancy is an essential property that artists must strive with in their works, that 3D designers must provide in their conceptions so that the human vision system perceives the object beauty. Similarity detection within 3D models is then a first step towards numerous interesting applications. In CAD, automatic search of similarity between CAD models is used primarily for model retrieval and indexing in large scale CAD databases [Car06a, Chu06a, Che12a, Liu13a]. In that context, end-users request automatic searches for "similar enough" designs according to a given model or sketch. Thus, the design reuse is encouraged by making use of existing components. For 3D meshes, many applications are studied such as pattern recognition, form editing or data completion. For example, Mitra et al. presented a symmetrization algorithm for geometric objects that enhances approximate symmetries of a model while minimally altering its shape [Mit07a]. Chaouch et al. [Cha08a] considered the reflection as the main characteristic to align their 3D models. Li et al. [Li11a] proposed a skull completion framework based on symmetry and surface matching. With the particular attractiveness of NURBS surfaces in 3D design industry, the similarity detection would certainly be useful. In fact, designers rarely start their works from scratch, but rather adapt existing models to meet new requirements. Statistically, it is shown that more than 75% of design activity involves reusing existing designs or starting from existing designs to address new designs [Iye05a]. Besides, parametric NURBS representations allow to easily and reliably access differential informations over the surfaces. Their representation by control points also gives the designer intuitive control. Hence, local similarities detection should be interesting for *reverse engineering*, allowing in one hand the analysis of a given 3D model, and in the other hand shape editing that is coherent with the detected similarities. Data compression in order to limit the storage size of a model can also benefit from the redundancy identified in similar parts. As far as we know, no research so far was dedicated to detecting the local similarities on parametric models like B-Spline or NURBS based B-reps. This article presents a method allowing the identification of NURBS surface patches that are similar to an approximated isometry. Our contributions are as follow. First, to find the best orientation of vectors of the characterized local frame at a point on the surface, we propose a simple method by analysing neighbourhood properties. We thus distinguish between direct and indirect isometries and propose to partition the isometries into five subsets. This classification simplifies the clustering and improves the identification of isometries. We further improve the clustering step by applying a spectral clustering algorithm. Unlike Mean Shift algorithm, our approach is fully unsupervised, and as such, is able to group automatically clusters without customizing global parameters. The remainder of this article is organized as follow. Section 2 reviews some previous works and our approach in this work. Section 3 describes the proposed pipeline of our algorithm that is detailed in the following sections. Section 8 shows some results of similarity detection among numerous experiments. Section 9 presents our conclusion and future works.

2 PREVIOUS WORK

In recent years, many articles have been published on similarity detection both in 2D image processing and in 3D modeling. In a first approach, Zabrodsky et al. [Zab95a] quantified existing symmetries within 2D and 3D objects, using a metric called the *symmetry distance*. The symmetry distance of a shape is defined to be the minimum mean squared distance required to move points of the original shape in order to obtain a symmetrical shape. Sun et al. [Sun97a] converted the symmetry detection problem into the correlation of Gaussian images; rotational and bilateral symmetries are identified by applying orientation histograms.

For 3D shape matching, two dominant techniques were proposed. First, global feature-based techniques represents 3D objects as a set of global features, for example, spherical-kernel moments [Cyb97a], or reduced feature vectors [Car06a]. The other set of methods uses graph-based techniques: the solid models are converted into attributed graphs that represent the geometrical and topological relationship between models entities [Hil01a, Ma10a]. However, in both cases, these techniques can neither identify similar parts within a model nor compute the transformation between these similar parts. Recently, many papers proposed to identify similarities within 3D meshes [Kaz04a, Pod06a, Ber08a, Bok09a, Lip09a, Mit13a] with different approaches like planar-reflective symmetry, graph-based matching, or votes transformation space. Kazhdan et al. [Kaz04a] introduced a *reflective symmetry descriptor* that represents a measure of reflective symmetry for an arbitrary 3D model for all planes through the model's center of mass. Podolak et al. [Pod06a] generalized this approach to identify symmetries of 3D objects associated with an arbitrary plane. Graph-based approach requires detecting local features on 3D shape from which a neighborhood graph is build to describe the coarse scale similarity structure of the object. Berner et al. [Ber08a] perform subgraph matching in graphs of feature points while Bokeloh et al. [Bok09a] apply feature lines.

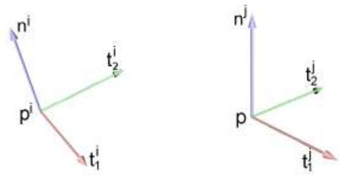


Figure 1: Local Frames of two similar points p_i et p_j according to right hand rule.

Other recent works [Lip09a, Mit13a] applied new technique in symmetry detection that we call *votes transformation space*. This technique bears some similarity to the Hough transform: points on the model with similar features are paired. A points pair corresponds to the transformation between the two points and their features; these transformations are cast to the transformation space and form a constellation of transformation votes. Clusters of these votes are candidates for defining similar parts in the model. While Mitra et al. [Mit13a] use Euclidean transformations as the feature to extract similarity, Lipman et al. [Lip09a] adopt Mobius transformations.

Among these approaches, the votes transformation space attracts our interest since it allows to retrieve a large class of potential transformations and it is able to identify similar parts in existing 3D objects and to characterize the transformation. In order to give a general view of this scheme, we detail the algorithm proposed in [Mit13a] that consists in the following four steps:

1. *Sampling and analysis:* a set of points is sampled over the surface of a 3D object. Since point positions are not sufficient to determine a general Euclidean transformation, geometry features at each sample are computed (the principal curvatures and a local frame composed of the principal directions and a normal vector). The signature is the couple of principal curvatures; points on the surface are paired if they have the same signature.
2. *Pairing:* each pair of points is associated a transformation corresponding to a vote in transformation space. Given two points p_i and p_j with their local

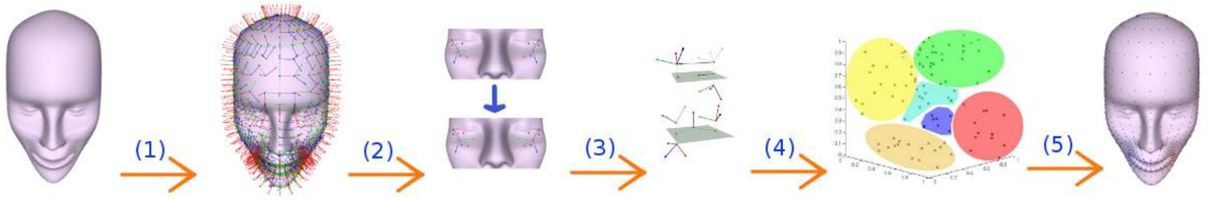


Figure 2: Proposed pipeline – (1) Sampling and signature computation, (2) Pairing and orientation, (3) Classification of isometry, (4) Clustering, (5) Validation.

(orthonormal) frames consisting in two tangents and a normal (figure 1), the transformation T_{ij} is computed so that p_i and its frame are mapped into p_j and p_j's frame. This transformation is then cast into votes of transformation space Γ.

3. *Clustering:* in transformation space Γ, each point T_{ij} represents a transformation between two similar points. Hence, clusters of similar transformations are identified since they may characterize two similar parts of the object.
4. *Patching:* ideally, a cluster of the previous step is a set of point pairs which belong to a couple of surface patches similar up to a transformation close to the cluster. However, spatial coherence between point pairs is lost in transformation space. Thus, this step enforces spatial coherence of the point pairs by applying an incremental region growing algorithm.

Our proposed pipeline follows the same votes transformation space approach. Our contributions are as follows. First, to find the best vectors orientations of the characterized local frame at a point on the surface, we propose a simple method by analysing neighbourhood properties. We thus distinguish between direct and indirect isometries and propose a partition the isometries into five subsets. This classification simplifies the clustering and improves the identification of isometries. We further improve the clustering step by applying a spectral clustering algorithm. Unlike Mean Shift algorithm, our approach is fully unsupervised, and as such, is able to group automatically clusters without customizing global parameters. In the following section, we described our isometry detection relative to these four steps.

3 PROPOSED PIPELINE FOR ISOMETRY DETECTION

Our work aims at identifying surface patches in a B-rep model that are similar up to an approximated isometry (we do not consider scaling). To identify the similarities, we adapt *the votes transformation space* that are used successfully in 3D meshes area [Lip09a, Mit13a]. Our pipeline consists in five consecutive steps. First, points are sampled over all B-reps of a CAD model by a sampling technique that adapts the parameterization (section 4.3). When the signature at each point is computed, vector directions are determined by parameterization, so it is not a geometric property of the surface. For this reason, local frames are not coherent, in particular to identify indirect isometries. We propose then a simple method to overcome this problem (section 4.4). Isometries between pairs of points are computed and partitioned into five types, based on orientation and on their fixed points (section 5). Next, clustering is applied in these five different spaces using a fully unsupervised spectral clustering algorithm to extract the evidence of existing similarity in the model (parameters are automatically computed). The isometries classification has two advantages: first it simplifies the clustering, but it also maps the pairs in transformation spaces of reduced dimensions. In this pipeline, the computation of the transformations is a major concern that affects considerably the quality of the result. By parametrizing the isometries differently, we improves the identification of isometries. Finally, similarities among local patches are identified following an adaptive growing process adapted for multiple faces in B-rep models (section 7).

4 COMPUTATION OF THE SIGNATURES

In our setting, we work with B-rep models based on trimmed free-form patches made of NURBS tensor product surfaces. For the first three steps of the similarity detection pipeline, it is sufficient to consider the patches independently. Thus, in this section, we focus on NURBS tensor product surfaces and in particular in computing a set of sample points and their characterizing signatures.

4.1 NURBS based models

Let S be a tensor product NURBS surface of bi-degree (p,q) associated to two knots vectors $\mathbf{u} = \{u_0,\ldots,u_n\}$ and $\mathbf{v} = \{v_0,\ldots,v_m\}$ and a set of control points $C = \{P_{ij} \mid i \in [0,n-p], j \in [0,m-q]\}$ weighted by $w_{ij} \in \mathbb{R}$, defined by the following equation:

$$S(u,v) = \frac{\sum_{i=0}^{n-p}\sum_{j=0}^{m-q} N_{i,p}(u)N_{j,q}(v)w_{ij}P_{ij}}{\sum_{i=0}^{n-p}\sum_{j=0}^{m-q} N_{i,p}(u)N_{j,q}(v)w_{ij}}. \qquad (1)$$

In a B-rep model, faces are not only represented by this type of NURBS, but also by other types such as planes, cylinders or spheres. However, one of the advantages of NURBS is that we can represent free-form as well as quadric surfaces [Cui11a].

4.2 Local differential properties: computation of the signature

Any point on the parametric surface, corresponding to a parametric coordinates (u, v), is attached to a set of persistent properties which is called *the signature* at that point. In our work, the signature at each point is composed of the two principal curvatures and an *orthonormal affine frame* having origin at that point, the unit vectors are the normal vector and the two principal directions (i.e. tangent vectors associated to the considered principal curvatures). The signature computation at a specific point on NURBS surface is based on local differential properties that could be evaluated from *the first and the second fundamental form* [Str61a, Far92a]. The *first fundamental form* that describes completely the metric properties of a surface, is defined as the distance of two points on a curve of the surface:

$$ds^2 = E\,du^2 \;+\; 2F\,du\,dv \;+\; G\,dv^2 \tag{2}$$

where $E = S_u \cdot S_u$, $F = S_u \cdot S_v$, $G = S_v \cdot S_v$, and ds is also called *the element of arc*.

The first fundamental form states that, for a given point p, partial derivatives S_u and S_v generate a tangent plane to the surface of origin p. Hence, the unitary normal vector is:

$$n = \frac{S_u \wedge S_v}{\left\|S_u \wedge S_v\right\|} = \frac{1}{\sqrt{EG - F^2}}\,(S_u \wedge S_v) \tag{3}$$

It associates to non normalized vectors S_u, S_v to form an affine frame of origin p.

Next, *the second fundamental form* of a parametric surface is defined by:

$$\kappa \cos\phi\, ds^2 = Ldu^2 + 2Mdudv + Ndv^2 \tag{4}$$

where $L = S_{uu} \cdot n$, $M = S_{uv} \cdot n$, $N = S_{vv} \cdot n$, and S_{uu}, S_{uv}, S_{vv} are second partial derivatives at p.

Equation (4) means that, for a given direction du/dv in u, v plane and a given angle ϕ, the *first and second fundamental forms* allow us to compute the curvature κ of a curve traced on the surface, also the tangent pointing toward this direction.

For this reason, two symmetric matrices are introduced:

$$\mathscr{F}_1 = \begin{pmatrix} E & F \\ F & G \end{pmatrix} \; and \; \mathscr{F}_2 = \begin{pmatrix} L & M \\ M & N \end{pmatrix} \tag{5}$$

Because S_u and S_v are linearly independent, $\mathscr{F}_1$ is always invertible. The matrix $\mathscr{F}_1^{-1}\mathscr{F}_2$ is also symmetric and so always has real eigenvalues and orthogonal eigenvectors. As a result, the two eigenvalues κ_1, κ_2 are the two *principal curvatures* and the two eigenvectors $t_1 = (\xi_1, \eta_1)^T$, $t_2 = (\xi_2, \eta_2)^T$ define the two *principal directions*:

$$\begin{aligned} t_1 = &\quad \xi_1 S_u + \eta_1 S_v \\ t_2 = &\quad \xi_2 S_u + \eta_2 S_v \end{aligned} \tag{6}$$

As for umbilical points ($\kappa_1 = \kappa_2$), principal directions are not uniquely defined, thus we do not consider them. For other points, the orientation of t_1 and t_2 depends on the parameterization. Section 4.4 details the way we orient the frame vectors.

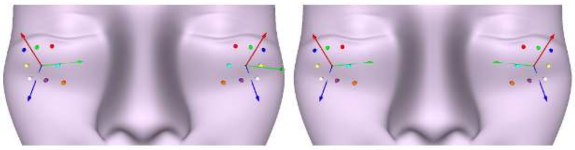


Figure 3: On the left: the orientation of the frame vectors follows the parameterization, so the two frames are not symmetric. On the right: we propose to find a coherent orientation of the vector frames by analyzing the points neighbors. Now, the two frames are symmetric, as is the underlying surface.

4.3 Sampling

Every point on the surface that is associated to a signature characterized by its local differential properties, might be potentially sampled for later computations. By benefiting from the facilities offered by parametric surfaces, a net of sample points on the surface is obtained by sampling uniformly the two parameters u and v (see equation 1). However, the parameterizations between surfaces in B-rep models vary. The uniform sampling along u and v may lead to a sparse net of sample points (figure 4a). To have a relatively uniform distance between points among all surfaces, we propose an iterator method to determine the two parameter gaps based on the distance between two points on each surface (figure 4b). In addition, the sampling affects the following steps of the algorithm in two ways. First, the denser sampling is, the better result is. Second, the denser samples also worsen the performance. For this reason, we evaluate a net of points uniformly on the surface but select randomly a limited number of samples following a uniform law on this points net (figure 4c). Moreover, the initial samples net is reserved for the validation step.

4.4 Robust surface orientation

Two sample points p_i and p_j are considered similar if their principal curvature matches, that is, $\kappa_1^i \sim \kappa_1^j$ and $\kappa_2^i \sim \kappa_2^j$. Two similar points are paired to evaluate the transformation between them. As mentionned in section 3, the orientation of the local frames vectors depend on parameterization. However, a coherent orien-

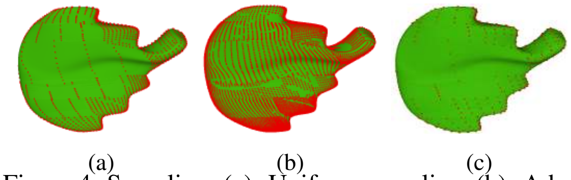
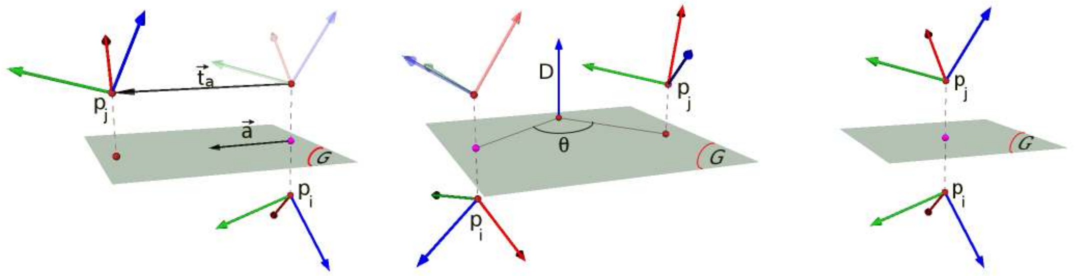


Figure 4: Sampling. (a): Uniform sampling, (b): Adaptive sampling, (c): Chosen sample points



(a) No fixed point (b) Line of fixed points (c) Plane of fixed points

Figure 5: Classification of Indirect Isometries based on fixed points. Here, local frames consist of a normal (red vector) and two principal directions (blue and green vectors).

tation of the frame is necessary, for example to distinguish direct from indirect transformation. The normal vector well oriented (and coherently for the whole surface) by the parameterization, but we modify the direction of tangent frame vectors. For each pair (p_i, p_j), we identify the orientation of principal vectors at p_j that is the most coherent to direction associated to those at p_i. Suppose that the frame at point p_i is fixed, in other words, the direction of vectors t_1^i and t_2^i is arbitrarily fixed. Consider now the frame at p_j. Each of the tangent vector at p_j can be oriented arbitrarily. Considering both tangent vectors, there are four possible different orientations of principal vectors at p_j.

We project the neighbours of p_i into the tangent plane, and order them into a sequence by turning around p_i. This gives us a reference list of curvatures. The four lists of neighbours of p_j corresponding to the four possible orientations of t_1^j and t_2^j are compared to the reference list. The chosen directions are thus the one that minimizes the sum of squares of differences between its list and the reference list.

Figure 3 shows a case of a plane symmetry where the initial orientation of vectors would have led to identifying a (wrong) direct transformation between points p_i and p_j.

5 ISOMETRY SPACES

Instead of considering all transformations in a 6-dimensional transformation space [Mit13a], we first partition the isometries and map them into one of the five isometry spaces. The advantage of these classifications is two fold: it simplifies the clustering, but also, it expresses the transformation in a space with the appropriate dimension. As an example, clustering translations in the original 6-dimensional transformation space requires the clustering algorithm to discriminate between points that belong to a degenerated 3-dimensional subspace. In our approach, the clustering will be applied directly in this subspace, taking into account only the relevant parameters.

5.1 Computation of the isometry

Given a points pair (p_i, p_j) as in the figure 1, we would like to evaluate the transformation from p_i to p_j so that p_i move to p_j's position and that the computed orthonormal frame at p_i aligns to the frame at p_j. We denote R_{ij} the rotation between these two frames and t_{ij} the corresponding translation. The computation is as follow:

$$R_{ij} = \begin{bmatrix} n^i \\ t_1^i \\ t_2^i \end{bmatrix}^T * \begin{bmatrix} n^i \cdot n^j & n^i \cdot t_1^j & n^i \cdot t_2^j \\ t_1^i \cdot n^j & t_1^i \cdot t_1^j & t_1^i \cdot t_2^j \\ t_2^i \cdot n_j & t_2^i \cdot t_1^j & t_2^i \cdot t_2^j \end{bmatrix} * \begin{bmatrix} n^i \\ t_1^i \\ t_2^i \end{bmatrix} \quad (7)$$

$$t_{ij} = p_j - R_{ij} * p_i \quad (8)$$

The transformation R_{ij} is an orthogonal matrix, i.e. $R_{ij} \in O(3)$, thus $T_{ij} : p_i(n^i, t_1^i, t_2^i) \longmapsto p_j(n^j, t_1^j, t_2^j)$ is then an isometry. Hence, T_{ij} belongs to $Is(X)$, the isometry group. We denote $\overrightarrow{T_{ij}}$ the associated linear transform, that is, the transform of matrix R_{ij}.

5.2 Classification of isometries

Affine isometry in three dimensional space, can be classified by considering the nature of its fixed points, according to the following theorem [Tis88a].

Theorem 1 *Given $T \in Is(X)$, there exists a unique couple $(g, t_{\overrightarrow{a}})$ where g is an isometry having a non empty set of fixed points G and here $t_{\overrightarrow{a}}$ is a translation of $\overrightarrow{a} \in \overrightarrow{G}$ such that $T = t_{\overrightarrow{a}} \circ g$. Additionally:*

- *$T = g \circ t_{\overrightarrow{a}}$ and $\overrightarrow{G} = E(1, \overrightarrow{T})$, the vector subspace associated with the eigenvalue* 1.
- *$T = g$ and $\overrightarrow{a} = 0$ if and only if T has at least one fixed point.*
- *If T has no fixed point, $dimG \geq 1$.*

In our case, suppose that $\overrightarrow{T}$ is not the identity and $\alpha = dimE(1, \overrightarrow{T})$, $\overrightarrow{T}$ is direct if $det(\overrightarrow{T}) = 1$ and $\overrightarrow{T}$ is indirect if $det(\overrightarrow{T}) = -1$. We can deduce the isometry type of T depending on its fixed points, as follow:

Direct Isometry

1. *A line (D) of fixed points* $(\alpha = 1, \overrightarrow{a} = 0)$: T is a rotation around the line (D) directed by $\overrightarrow{n} \in E(1, \overrightarrow{T})$.

2. *No fixed point* ($\overrightarrow{d} \neq 0$): T is either a translation of $\overrightarrow{d}$ or the composition of a rotation around (D) directed by $\overrightarrow{d}$ and a non-zero translation colinear to (D).

Indirect Isometry

1. *A unique fixed point* A ($\alpha = 0$, $\overrightarrow{d} = 0$): T consists of a rotation around an axis (D) directed by $\overrightarrow{n} \in E(-1, \overrightarrow{T})$ and passing through A, and a reflection relative to the plane (G) containing A and perpendicular to (D) (figure 5a).
2. *A plane G of fixed points* ($\alpha = 2$, $\overrightarrow{d} = 0$): T is a symmetry relative to the plane G that is defined by $\overrightarrow{n}_{1,2} \in E(1, \overrightarrow{T})$ (figure 5b).
3. *No fixed point* ($\overrightarrow{d} \neq 0$): T is composed of a symmetry relative to a plane G whose the normal $\overrightarrow{n} \in E(-1, \overrightarrow{T})$, and a non-zero translation parallel to this plane (figure 5c).

Table 1 details the classification of isometries into five subsets. These groups will be treated separately to detect similar patches either among these surfaces or in a surface itself.

While the groups of direct isometries identify approximated patches by rotating and/or translating, the group of indirect isometries determine approximated ones by reflecting.

Iso FP	Indirect	Direct
Without	$T_{ij} = s_G \circ t_{\overrightarrow{d}}$	$T_{ij} = r(D, \theta) \circ t_{\overrightarrow{d}}$
Line of	$T_{ij} = s_G \circ r(D, \theta)$	$T_{ij} = r(D, \theta)$
Plane of	$T_{ij} = s_G$	Not possible

Table 1: Classification of the isometries based on isometry types (Iso) and nature of fixed points (FP); s_G is a symmetry relative to the plane G; $t_{\overrightarrow{d}}$ is a translation of vector $\overrightarrow{d}$; $r(D, \theta)$ is a rotation of angle θ around axis (D).

5.3 Comparison of two isometries

We now have five different transformation spaces, and for each, will apply a clustering algorithm. The clustering need to have a distance in each of these spaces, that is, we derive distances for two isometries of the same type.

For direct isometries, the components of isometries are the rotation axis (D) and angle θ, and the translation $t_{\overrightarrow{d}}$. As the rotation axis and the translation have the same direction, the translation vector $\overrightarrow{d}$ and a point P on the axis are sufficient. For comparing the rotations we use the angles and the distance between the two axes, and the difference of the angles; for translations, we still compare the length of the translation vectors (the angle is the same as for the axes).

For indirect isometries, the analysis is identical to the direct setting, except for the symmetry plane G. The comparison between planes consists in comparing the normals to these planes and computing the distance between the mid-point and the plane.

In the following, we denote $d(T, T')$ the distance between the two isometries T and T' corresponding to the two point pairs (p_i, p_j) and $(p_{i'}, p_{j'})$; M_{ij}, $M_{i'j'}$ the midpoints of $[p_i, p_j]$ and $[p_{i'}, p_{j'}]$; $dist(P, G)$ denotes the distance from a point, line or a plane to another one.

Direct isometries

$$\begin{aligned} d(T, T') = \quad & (1 - |cos(D \cdot D'|) + \frac{|(\theta - \theta')|}{2\pi} \\ & + \omega_1 dist(D, D') + \omega_2 |(\|t\| - \|t'\|)| \end{aligned} \tag{9}$$

Indirect isometries

$$\begin{aligned} d(T, T') = \quad & (1 - |cos(\overrightarrow{n} \cdot \overrightarrow{n'})|) \\ & + \omega_1 (dist(M_{ij}, G') + dist(M_{i'j'}, G)) \\ & + \frac{|(\theta_{ij} - \theta_{i'j'})|}{\pi} \end{aligned} \tag{10}$$

The weight ω_i are chosen as the diagonal of the bounding box of the model and so that the terms all vary between 0 and 1.

6 CLUSTERING

After computing the isometries as described in the previous section (Section 5), the clustering step aims at grouping pairs of points having approximatively the same isometry. This step is based on a spectral approach called spectral clustering and differs from the Mean Shift algorithm [Mit13a] which requires difficult parameters tuning.

6.1 Method

Introduced in machine learning by Shi et al. [Shi00a, Von07a], the spectral clustering is an unsupervised method that consists in extracting dominant eigenvectors of a *normalized Gaussian affinity matrix*. These eigenvectors span a low dimensional spectral embedding in which projected data are grouped into clusters. We describe the different steps of this clustering method below.

Let $d(T, T')$ be the distance between the two isometries T and T' in the same class corresponding to the two point pairs (p_i, p_j) and $(p_{i'}, p_{j'})$. Note that $d(T, T')$, and consequently the affinity measure (11), will change depending on the class of the isometries (as described in section 5.2). Let $S = \{(p_i, p_j)\}_{\{i,j=1..N_l\}} \in \Gamma_l$, $l \in [|1, 5|]\}$ be the set of N_l pairs in the l-th isometry space and let k be the number of clusters.

This method is based on Gaussian affinity measure, its parameter and their spectral elements. It uses inherent properties of the Mercer kernel (here, the Gaussian kernel) that implicitly projects data in a large scale dimension space where data will be linearly separable. In other words, the Gaussian measure defines a nearness criterion in the linear vector space and weights the

```
SpectralClustering $(S,k)$
Construct the affinity matrix $A \in \mathbb{R}^{N_l \times N_l}$ defined by:
    $A_{ii'} = \begin{cases} e^{\left(-\frac{d(T,T')^2}{(\sigma/2)^2}\right)} & \text{if } (p_i,p_j) \neq (p_{i'},p_{j'}), \\ 0 & \text{otherwise.} \end{cases}$    (11)
Construct the normalized matrix : $L = D^{-\frac{1}{2}}AD^{-\frac{1}{2}}$ with $D_{i,i} = \sum_{r=1}^{N_l} A_{ir}, \forall i \in \{1,..N_l\}$.
Construct the matrix $X = [X_1X_2..X_k] \in \mathbb{R}^{N_l \times k}$ by stacking the $k$ largest eigenvectors of $L$.
Construct the matrix $Y$ by normalizing rows from matrix $X$.
Consider each row of matrix $Y$ as a point in $\mathbb{R}^k$ and group them into $k$ clusters with K-means method.
Assign the original point pair $(p_i,p_j)$ to class $\theta$ if and only if the $i^{th}$ row of matrix $Y$ is assigned to class $\theta$.
```

Algorithm 1: Spectral Clustering

matching scores. Moreover, classes of arbitrary shapes (in particular, non convex) may be defined [Von07a]. Furthermore, this algorithm only depends on two parameters which are the Gaussian Affinity parameter and the number of classes k. To make this method fully unsupervised, we adopt a heuristic to determine each parameter [Mou11a].

6.2 Affinity parameter

The expression of the Gaussian affinity, defined in equation (11), depends on the parameter σ. The parameter σ defines a threshold on transformation distances between point pairs (p_i,p_j). To set it, we consider a uniform distribution of the points, that is, such that all points are equidistant from each other.

Elements of S which defines an isotropic distribution are included in a bounding box of size $D_{max} = \max_{(p_i,p_j)\neq(p_{i'},p_{j'})} d(T,T')$ in each of the m dimension – $d(T,T')$ is defined in section 5.3. By dividing this box into N_l identical volumes, the (uniform) distance between two points is, noted D_{unif}, is:

$$D_{unif} = \frac{max_{(p_i,p_j)\neq(p_{i'}p_{j'})} d(T,T')}{N_l^{1/m}}. \tag{12}$$

where m is the dimension of the isometry space (varies depending on the nature of the isometry). We can consider that if a cluster exists, there are points that are separated by a distance lower than D_{unif}. Similarly, the Gaussian parameter σ is used as a fraction of distance $D_{unif} : \sigma = D_{unif}/2$. Thus this heuristic integrates a notion of density of points in a m-dimensional space, and derives a threshold from which points are considered closed.

6.3 Number of clusters

The choice of number of clusters is a general problem for all unsupervised clustering algorithms[Von07a]. To determine this number of clusters k, we adopt a try-and-test approach by exploiting the Gaussian affinity matrix A and defining a quality measure based on the ratio of Frobenius norms. Let α_k be a bound on the number of clusters to identify. For a value $k' \in [|2,\alpha_k|]$, the affinity matrix is indexed per cluster. A block matrix is then defined: off-diagonal blocks represent the affinity between clusters and diagonal blocks represent the affinity within the cluster. From this block structure, we can evaluate a mean ratio, called $r_{k'}$, between all off-diagonal blocs and the diagonal blocks in Frobenius norm. From this, among the values $k' \in [|2,\alpha_k|]$, the minimum of the ratio $r_{k'}$ defines the optimal number of classes k:

$$k = \arg \min_{k' \in [|2,\alpha_k|]} r_{k'}. \tag{13}$$

This minimum corresponds to a case where the affinity between clusters is the lowest and the affinity within cluster is the highest. More details in this interpretation can be found in [Mou11a].

7 VALIDATION

Ideally, every class obtained by the clustering is a set of point pairs which belong to a couple of surface patches similar up to an approximated isometry. However, spatial coherence between point pairs is lost during the isometry clustering. Therefore, the purpose of the *validation* is to overcome this problem in order to identify similar patches. We present the validation step within a NURBS patch (section 7.1) and then consider region growing over a B-Rep model, which may include multiple NURBS patches (section 7.2).

7.1 Validation within a NURBS patch

The validation is performed by a region expanding process. Given C_k, a class of points pairs in an isometry space, a pair (p_i,p_j) is selected randomly. The chosen isometry T_{ij} is applied to the eight neighbours of p_i, their images are thus compared to eight neighbours of p_j. If the deviation of any neighbour is under a chosen threshold, the points pair is accepted as belonging to the two similar patches. This process continues iteratively; we further test the neighbours of p_i. It is repeated until all points on the surface are visited, or the condition does not hold, or until all pairs in class are considered. This step generates a candidate for two similar patches. Nevertheless, this process stops at the boundary of the surface. But a 3D object modelled by NURBS based B-rep is composed by several NURBS surfaces.

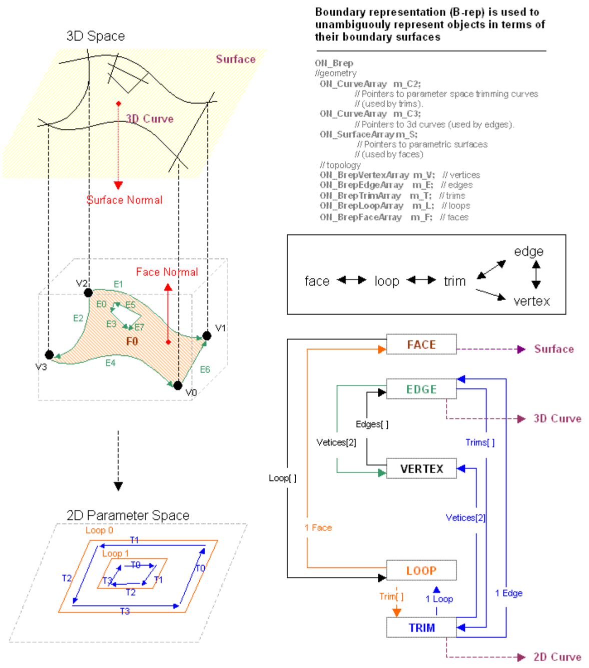
7.2 Validation within a B-rep

The figure 6 represents an overview of the B-rep specification in the context of OpenNURBS. In fact, a NURBS based B-rep is a set of trimmed NURBS that consists in a surface and some trimming contours. The trimming

```
Find the closest edge $e$ to $p$
if $e$ is shared with other face then
    Determine the adjacent face $S$
    Take the set $P$ of points on all edges of $S$
    Find $q$ the closed point to $p$ in $P$
    Find curvilinear parameters of $q$
```

Algorithm 2: Identification of adjacent point

contours define which parts of the surface are kept or removed. In OpenNURBS context, the *loop* is an abstraction of a trimming contour. It is defined by a set of closed trimming curves that are in turn expressed by *trims*. Each *trim* is attached to a 2D curve and an *edge*. The 2D curve defines the curvilinear coordinates of the *trim* within the surface. The *edge* is a 3D curve on the surface and is a boundary. Furthermore, an edge can be shared among multiple trims. Given p the point on the boundary of the surface where the validation cannot continue. The proposed algorithm 2 identifies a point q on an adjacent surface close to p.


Boundary representation (B-rep) is used to unambiguouly represent objects in terms of their boundary surfaces

```
ON_Brep
//geometry
  ON_CurveArray  m_C2;
            // Pointers to parameter space trimming curves
            // (used by trims).
  ON_CurveArray  m_C3;
            // Pointers to 3d curves (used by edges).
  ON_SurfaceArray m_S;
            // Pointers to parametric surfaces
            // (used by faces)
  // topology
  ON_BrepVertexArray m_V;   // vertices
  ON_BrepEdgeArray   m_E;   // edges
  ON_BrepTrimArray   m_T;   // trims
  ON_BrepLoopArray   m_L;   // loops
  ON_BrepFaceArray   m_F;   // faces
```

Figure 6: Boundary representation (B-rep) in the context of OpenNURBS (from http://wiki.mcneel.com/).

8 EXPERIMENTS

We have implemented the pipeline described in section 3 to identify the similar patches within the following B-Rep models. We use CAD models under OpenNURBS specifications (http://www.opennurbs.org/) for our experiments. In general, the main tool that affects directly on the robustness of our pipeline is the *surface orientation* algorithm (section 4.4) and the classification of isometries (section 5.2). In the following, we propose some test scenarios to validate these tools following by the results on some CAD models of our pipeline.

Since the *surface orientation* algorithm is only applicable for indirect isometries, the models for our test cases exhibit only these types. We proposed three B-rep models of leaves as showed in the figure 7. Given N_{Exp} the number of *expected re-oriented* pairs and N_{Total} the *total number of pairs* computed in each model. Then, the tolerance rate R_{Tol} is the ratio between these two factors.

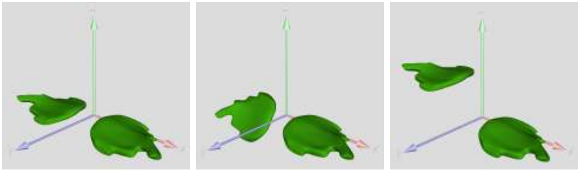
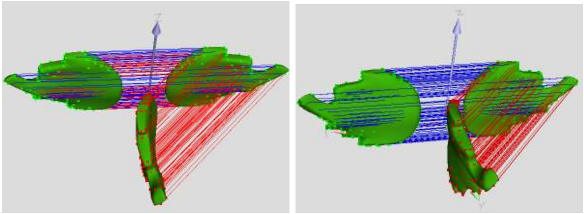

(a) Sym. (b) Sym. + Rot. (c) Sym. + Trans.

Figure 7: Proposed CAD models representing the symmetry (Sym.), the rotation (Rot.) and the translation (Trans.) for the *surface orientation* algorithm test cases.

According to table 2, our test cases shows that this algorithm has a tolerance rate up to 80%. Despite the orientation still failed at points whose the opposite neighbors (symmetric via these points in the parameters net) are similar, this algorithm guarantees that the *classification of isometries* is reliable and thus the *similarity detection* is robust. Next, by applying our algorithm of *Automatic Spectral clustering* [Mou11a], the results of clustering in the figure 8 illustrate the effectiveness between *Euclidean transformation* approach [Mit13a] and our approach of *classification of Isometry*. This figure represents three leaves in a model that have two symmetric pairs of leaves. Besides, every line that connects two points having the same signature corresponds to a point in transformation space. Additionally, lines with the same color are in the same cluster (i.e. the same transformation in general). In this test case, we use the computation of Euclidean transformation and the distance metric as represented by Mitra et al. [Mit13a]. We can observe that while there are some wrong classified points in the *Euclidean transformation* approach (figure 8a), our approach can address this problem (8b). In other words, with the aid of the classification of isometries, the output of the clustering algorithm was significantly improved. Moreover, the use of *Automatic Spectral clustering* algorithm also contributes to the robustness of our pipeline. In fact, the results shown in this figure are obtained without tuning any parameter.

Model	N_{Exp}	N_{Total}	R_{Tol}
7a	520	621	0.84
7b	509	618	0.82
7c	505	621	0.81

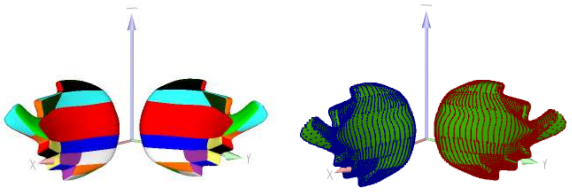
Table 2: Tolerance rate of the *surface orientation* algorithm.



(a) (b)

Figure 8: Comparison of the effectiveness between the *Euclidean transformation* approach 8a and the *classification of Isometry* approach 8b.

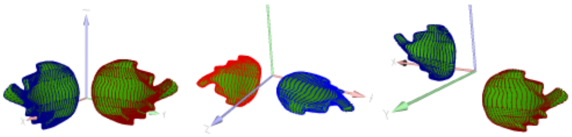
Also, the figure 9 presents the result of our proposed validation within B-rep. The figure 9a shows that there are two separated B-reps that are formed by several trimmed NURBS surfaces displayed by different colors. As in the figure 9b, the validation has successfully validated all the points over the surface of these B-rep objects.



(a) Original B-reps (b) Result

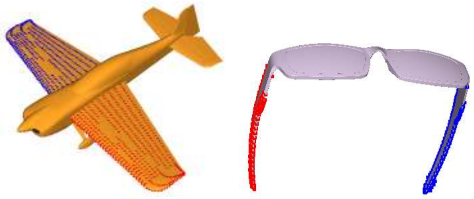
Figure 9: Result of validation within a B-rep.

Finally, the figures 10, 11, 12 and 13 represent the final results of our experiments on some CAD models downloaded from GrabCAD (http://grabcad.com/). These results represent different isometries detected by our proposed pipeline. The first set of leave models exhibit the indirect isometries. In fact, while the figure 10a shows a symmetry, the figure 10b represents a symmetry following by a rotation axis, and a symmetry following by a translation is detected in the figure 10c. Also, the figures 12a and 12b describe the direct isometries between the four legs of a dragon: this isometry is decomposed into a translation and a rotation. The figures of the plane and the sunglasses demonstrate the symmetry between different parts in these models. In addition, the figure 11a also demonstrates a direct isometry composed by a rotation axis between the two parts of the plane tail. Next, the figure 13a and 13b describe the similarity detection result of a series of human head in a model, in which, from left to right, every head presents a refinement step on the surface. In other words, there are some deformations between these heads. When applying our pipeline, one of the identified transformations is the translation between the green dots and the blue dots (figure 13a), another is the symmetry inside a B-rep (figure 13b). This result demonstrates that our pipeline works well even if there is a slight deformation between the similar surfaces.



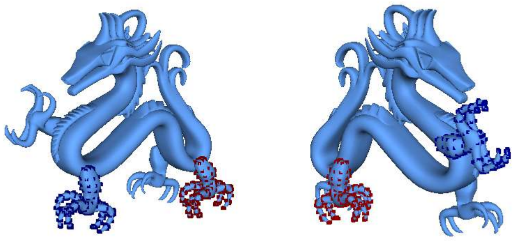
(a) (b) (c)

Figure 10: Similarities in leaves models.



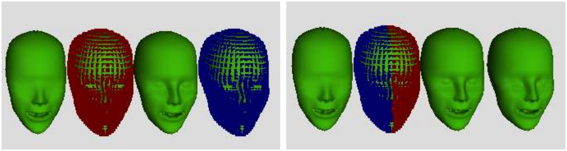
(a) (b)

Figure 11: Symmetry detected in models.



(a) (b)

Figure 12: Direct isometry detected in models.



(a) (b)

Figure 13: Similarities detected in a model of human heads.

9 CONCLUSION

In this article, we propose an algorithm to identify similar parts within objects modelled by NURBS based B-Reps, by adapting and improving the *votes transformation space* approach described by Mitra et al. [Mit13a]. Beside adapting the approach for parametric representations, we have proposed a local coherent frames orientation simply based on the points neighbours. A (robust) globally coherent orientation is then insured at the validation step. The local orientation allows to sort *direct* and *indirect isometries*. Furthermore, based on the analysis of fixed points, local isometries are further partitioned into five subsets. The experiments show that this classification before the clustering steps significantly improves the results. Furthermore, the clustering was further improved by using a fully unsupervised spectral method, for which, unlike for the *Mean-shift* algorithm, parameter tuning is not necessary. In particular, the number of isometries (clusters) to be identified does not need to be known in advance. Finally, the validation step extends the identified isometries among different NURBS patch within the B-rep. We are now able to recover isometric patches of B-splines or NURBS surfaces or similar to an isometry, or an approximate isometry (like shown in the experiment section). For future work, first we would like to filter the similarity detection by filtering similarities between control points. Second, we would like to exploit the isometries for applications: by linking the control structures corresponding to these patches, to offer the possibility to coherently edit or segment the models. Moreover, we could use the similarity to limit the storage size of the model.

10 REFERENCES

[Baj97a] Bajaj, C., Lee, H. Y., Merkert, R., and Pascucci, V. NURBS based B-rep models for macromolecules and their properties. Proc. of Symposium on Solid modeling and applications, p. 217-228, 1997.

[Ber08a] Berner, A., Bokeloh, M., Wand, M., Schilling, A., and Seidel, H. P. A graph-based approach to symmetry detection. Proc. of conf. on Point-Based Graphics, p. 1-8, 2008.

[Bok09a] Bokeloh, M., Berner, A., Wand, M., Seidel, H. P., and Schilling, A. Symmetry detection using feature lines. Computer Graphics Forum, p. 697-706, 2009.

[Car06a] Cardone, A., Gupta, S.K., Deshmukh, A., Karnik, M. Machining feature-based similarity assessment algorithms for prismatic machined parts. Computer-Aided Design (C A D) 38, p. 954-972, 2006.

[Cau03a] Caumon, G., Sword Jr, C. H., and Mallet, J. L. Constrained modifications of non-manifold b-reps. Proc. of the symposium on Solid modeling and applications, p. 310-315, 2003.

[Cha08a] Chaouch, M., and Verroust-Blondet, A. A novel method for alignment of 3D models. Shape Modeling Internaltional, p. 187-195, 2008.

[Che12a] Chen, X., Gao, S., Guo, S., and Bai, J. A flexible assembly retrieval approach for model reuse. Computer-Aided Design 44, p. 554-574, 2012.

[Chu06a] Chu, C.H., and Hsu, Y.C. Similarity assessment of 3D mechanical components for design reuse. Robotics and Computer-Integrated Manufacturing 22, p. 332-341, 2006.

[Cui11a] Cuilliere, J.C., François, V., Souaissa, K., Benamara, A., and BelHadjSalah, H. Automatic comparison and remeshing applied to CAD model modification.Computer-Aided Design 43, p. 1545-1560, 2011.

[Cyb97a] Cybenko, G., Bhasin, A., and Cohen, K.D. Pattern recognition of 3D CAD objects: Towards an electronic yellow pages of mechanical parts. International Journal of Smart Engineering System Design, no 1, p. 1-13, 1997.

[Dim99a] Dimas, E., and Briassoulis, D. 3D geometric modelling based on NURBS: a review. Advances in Engineering Software, no 9, p. 741-751, 1999.

[Far92a] Farin, G. Courbes et surfaces pour la CGAO - Conception Géométrique Assistée par Ordinateur. Masson, 1992.

[Hil01a] Hilaga, M., Shinagawa, Y., Kohmura, T. Topology matching for fully automatic similarity estimation of 3D shapes. Proc. on Computer graphics and interactive techniques, p. 203-212, 2001.

[Iye05a] Iyer, N., Jayanti, S., Lou, K., Kalyanaraman, Y., and Ramani, K. Three-dimensional shape searching: state-of-the-art review and future trends. Computer-Aided Design 37, p. 509-530, 2005.

[Li11a] Li, X., Yin, Z., Wei, L., Wan, S., Yu, W., and Li, M. Symmetry and template guided completion of damaged skulls. Computers and Graphics 35, p. 885-893, 2011.

[Lip09a] Lipman, Y., and Funkhouser, T. Möbius voting for surface correspondence. ACM Transactions on Graphics (TOG). ACM, p. 72, 2009.

[Liu13a] Liu, Y.J., Luo, X., Joneja, A., Ma, C. X., Fu, X. L., and Song, D. User-Adaptive Sketch-Based 3D CAD Model Retrieval, 2013.

[Kaz04a] Kazhdan, M., Chazelle, B., Dobkin, D., Funkhouser, T., and Rusinkiewicz, S. A reflective symmetry descriptor for 3D models. Algorithmica 38, p. 201-225, 2004.

[Ma10a] Ma, L., Huang, A., and Wang, Y. Automatic discovery of common design structures in CAD models. Computers and Graphics 34, p. 545-555, 2010.

[Mit07a] Mitra, N.J., Guibas, L.J., and Pauly, M. Symmetrization. ACM Transactions on Graphics (TOG) 26, p. 63, 2007.

[Mit13a] Mitra, N. J., Pauly, M., Wand, M., and Ceylan, D. Symmetry in 3d geometry: Extraction and applications. Computer Graphics Forum, 2013.

[Mou11a] Mouysset, S., Noailles, J., Ruiz, D., and Guivarch, R. On a strategy for spectral clustering with parallel computation. Proc. of VECPAR. p. 408-420, 2011.

[Shi00a] Shi, J., and Malik, J. Normalized cuts and image segmentation. IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 22, p. 888-905, 2000.

[Str61a] Struik, D. J. Lectures on classical differential geometry. Courier Dover Publications, 1961.

[Sun97a] Sun, C., and Sherrah, J. 3D symmetry detection using the extended Gaussian image. IEEE Trans. on Pattern Analysis and Machine Intelligence 19, p. 164-168, 1997.

[Tis88a] Tisseron, C. Géométries affine, projective et euclidienne. Hermann, 1988.

[Pod06a] Podolak, J., Shilane, P., Golovinskiy, A., Rusinkiewicz, S., and Funkhouser, T. A planar-reflective symmetry transform for 3D shapes. ACM Trans. on Graphics, p. 549-559, 2006.

[Von07a] Von Luxburg, U. A tutorial on spectral clustering. Statistics and computing 17, no 4, p. 395-416, 2007.

[Zab95a] Zabrodsky, H., Peleg, S., and Avnir, D. Symmetry as a continuous feature. IEEE Transactions on Pattern Analysis and Machine Intelligence 17, p. 1154-1166, 1995.